AMENDMENTS TO THE SPECIFICATION

Page 1, paragraph [0002] insert the following heading:

BACKGROUND

Page 1, insert the sub-heading at line 5 to:

1. Technical Field

Page 1, paragraph [0007],

The present invention relates to a distributed computer system and to a method of operating a distributed computer system.

Page 1, insert the sub-heading at line 8 to:

2. Related Art

Please amend the paragraph beginning at page 1, line 30 as follows:

The above relates to stand-along computers – an early example of distributed computing was implemented by researchers at Xerox PARC in the early 1980's. They developed software for calling modules on other computers so that the modules appeared to the programmer as if they were on the same computer as the program which called them. This software is described in the paper "Implementing Remote Procedure Calls", Birrell A., and Nelson, B., ACM Transactions on Computer Systems, Vol. 2, No. 1, February 1984, Pages 39-59. That paper explains that Remote Procedure Calls are achieved by the addition of a communications program, and client

and server 'stub' modules which together cause the selected procedure to be carried out on the remote computer, and return the result of the procedure to the application program. In writing a distributed application, a programmer prepares the user and server programs and one or more ~~on~~ interface modules (a list of procedure names together with the types of the parameters the procedure takes and the type of the result in produces). The user is provided with software which automatically generates the 'stub' modules on the basis of the interface module prepared by the programmer.

Please amend the paragraph beginning at page 3, line 25 as follows:

In order to alleviate this problem, Sharma Chakravarthy and others proposed the adoption of an idea previously used in some database management software. In a paper "ECA Rule Processing in Distributed and Heterogeneous Environments", in the proceedings of the International Symposium on Distributed Objects and Applications, 1999, they suggest that the operation of a distributed application might be altered at run-time by making its operation dependent on Event-Condition-Action rules stored in a database. These are said to confer an "active" capability on the distributed computer system described therein. The proposed system is a distributed computer system each computer of which has CORBA middleware installed on it. One computer within the distributed computer system is designated a Composite Event Detection and Rule execution (CEDaR) server computer which:

Please amend the paragraph beginning at page 4, line 14 as follows:

The action is normally the execution of a software component (or at least a procedure or method within that software component) on one of the other computers making up the distributed computer system. In an alternative design, the action can be the execution of software contained within a local software library on the CEDaR server (such a library can be updated with new software without having to stop the execution of the distributed application program).

Please amend the paragraph beginning at page 4, line 21 as follows:

To make the distributed application adaptable, updateable ECA rules are interpreted at runtime. Of course, there is little point in interpreting an updated ECA rule at run-time, unless the executing program adapts its execution at run-time in order to reflect the change that has occurred in the database. To meet this requirement, the CEDaR server and the other computers in the distributed computer system proposed by Chakravarthy uses OrbixWeb from Iona Technologies (a CORBA-compliant Object Request Broker which enables communication between components executing as part of the distributed application program). OrbixWeb offers a Dynamic Invocation Interface which allows a programmer to write the executing program in such a way that the method named in the action field of the ECA rule stored in the database is automatically selected at runtime. Furthermore, the ability to use events defined ~~whilst~~ while the application is running is provided by using the Java Reflection Application Programmer interface (API) provided as part of the Java programming language package provided by Sun Microsystems.

Please amend the paragraph beginning at page 5, line 1 as follows:

The reliance on a single server to evaluate rules, and to call condition evaluation and rule execution procedures means that Chakravarthy's distributed computer system is highly dependent on the performance of that single server and sends and receives an amount of traffic which grows with the size of the distributed application being run.

Page 5, line 5, insert the following heading:

BRIEF SUMMARY

Please amend the paragraph beginning at page 5, line 6 as follows:

According to a first aspect of the present invention, there is provided a distributed computer system comprising at least two interconnected computers, each of said computers storing:

Please amend the paragraph beginning at page 5, line 26 as follows:

By storing, at each of two or more computers within a distributed computer system, each of which computers executes one or more components forming part of a distributed application program running on said computer, one or more event reaction rules specifying calls to functionality within said component to be made in reaction to the receipt of an event message, and enabling a user to update the event reaction rules, a distributed computer system with robust and y t easily updateable operation is provided.

Please amend the paragraph beginning at page 6, line 18 as follows:

According to a second aspect of the present invention, there is provided a method of operating a distributed computer system comprising a plurality of interconnected computers, said method comprising operating each of said computers to:

Please amend the paragraph beginning at page 6, line 22 as follows:

i) execute one or more component processes which form part of a distributed program running on said distributed computer systems;

Page 6, line 31, insert the following heading:

BRIEF DESCRITPION OF THE DRAWINGS

Page 7, line 13, insert the following heading:

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

Please amend the paragraph beginning at page 7, line 23 as follows:

The call data recording system 2 is based on call data received from exchanges ~~10, 12~~ 50, 51 within the telecommunication network. Each of the exchanges is programmed to send the call data it collects through the Public Switch Telecommunication Network 15 (PSTN) via local area network 16 to call data recording computer 18. The call data recording computer 18 saves the call data in mass storage device 20. Billing event generation software is loaded from CD-ROM 17 on the computer 18. The call data may have a format similar to that shown in table 1 below:

Please amend the paragraph beginning at page 12, line 32 as follows:

This is followed by an optional condition element 313. The <condition> element 313 consists of the sub-elements <BooleanExpression>, <true> and <false>. The BooleanExpression element 315 provides the condition predicate that will be evaluated by the policy handling software obtained from the CD-ROM 27 or the CD-ROM 36. There is a number of things this string can include, which depend on the semantics of the Policy Handler class included within the policy handling software. First, the Boolean Expression may contain a string with a syntax recongnisable by the Policy Handler. The policy handler will read the string in and evaluate it as is. For instance, the string may be a predicate expressed directly in Java syntax (e.g., a>3 && b==4), or alternatively in a policy handler-specific way (e.g., a greaterThan 3 AND b equals 4). Using the Java syntax to express the condition is easer because the symbols >, ==, 3 and 4 are directly parsable and understood by Java per se as they are Java symbols and it is only the symbols a and b that need make sense in the policy handler, i.e., the policy handler needs to know what a and b mean. Using a policy handler-specific way to express the predicate complicates the operation of the policy handler software, since it is on the policy handler to parse and understand not only a and b but all other symbols, i.e., greaterThan, AND and equals. Second, the string may contain an identifier for a condition. In this case, the policy handler will use the identifier to invoke a condition evaluation method/program (either local or remote, depending on whether the condition checks the state in the local or some remote context) that returns true or false. Following this, there are the <true> and <false> elements which ~~both~~ each point

to set of rules 317 and 319 respectively to be executed provided the condition evaluates to true or false respectively.

Please amend the paragraph beginning at page 17, line 1 as follows:

An identifier (413, ~~417~~ 415) for each of the events handled or generated by the component is also included.

Please amend the paragraph beginning at page 24, line 27 as follows:

From the above description it will be seen how the first embodiment offers the run-time modification of a distributed application as seen in the prior-art. However, it will also be seen how a modifiable method call in a component is modified at run-time in dependence on a policy stored on the same machine as the component whose operation is modified (and that is true of each of a plurality of computers involved in running a distributed application program). This obviates the need for complex software to handle dynamic invocation of methods stored on another computer and hence reduces the cost of providing a run-time modifiable distributed computer system without sacrificing modifiability of the operation of the distributed computer system.

Page 26, line 1, insert the following heading:

WHAT IS CLAIMED IS: